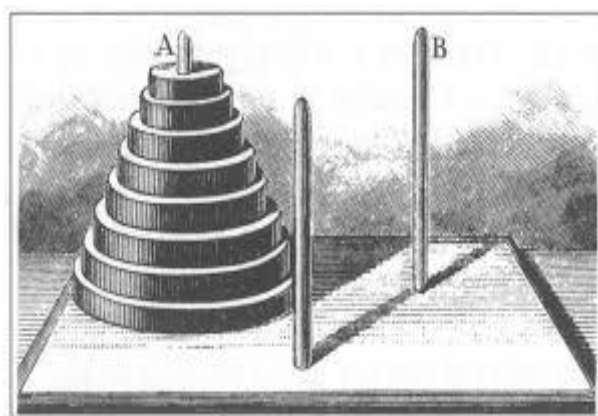


河內塔

河內塔，也被傳說稱之為梵天寺之塔問題，相傳很久以前在越南河內在這個問題中 64 個金盤的它每個的大小都是不一樣的，並在一開始的時候會依序從底部最大排到頂部最小，而搬移的規則有三個，分別是：

1. 一次只能搬移一個金盤
2. 盤子只能在三根柱子中被搬動（你不能拿起來放在旁邊，必須挑一根柱子放下）
3. 尺寸較小的盤子需永遠保持在最上方



塔中的僧侶們會依照一個古老的預言，並依規則來移動這些金盤；傳說中的 64 層就會需要 $2^{64} - 1$ 次，會需要超過 5849 億年才有辦法完成而預言則說當這些金盤全部都移動完畢之後，世界就會毀滅。

和 recursion 有什麼關係？

由上述的三個規則我們可以想到這種利用重複規則的問題在數學上我們會使用遞迴函數，而在我們撰寫的程式中，有重複利用自己執行性質的就屬 recursion 最適合。

怎麼用 recursion 解決此問題？

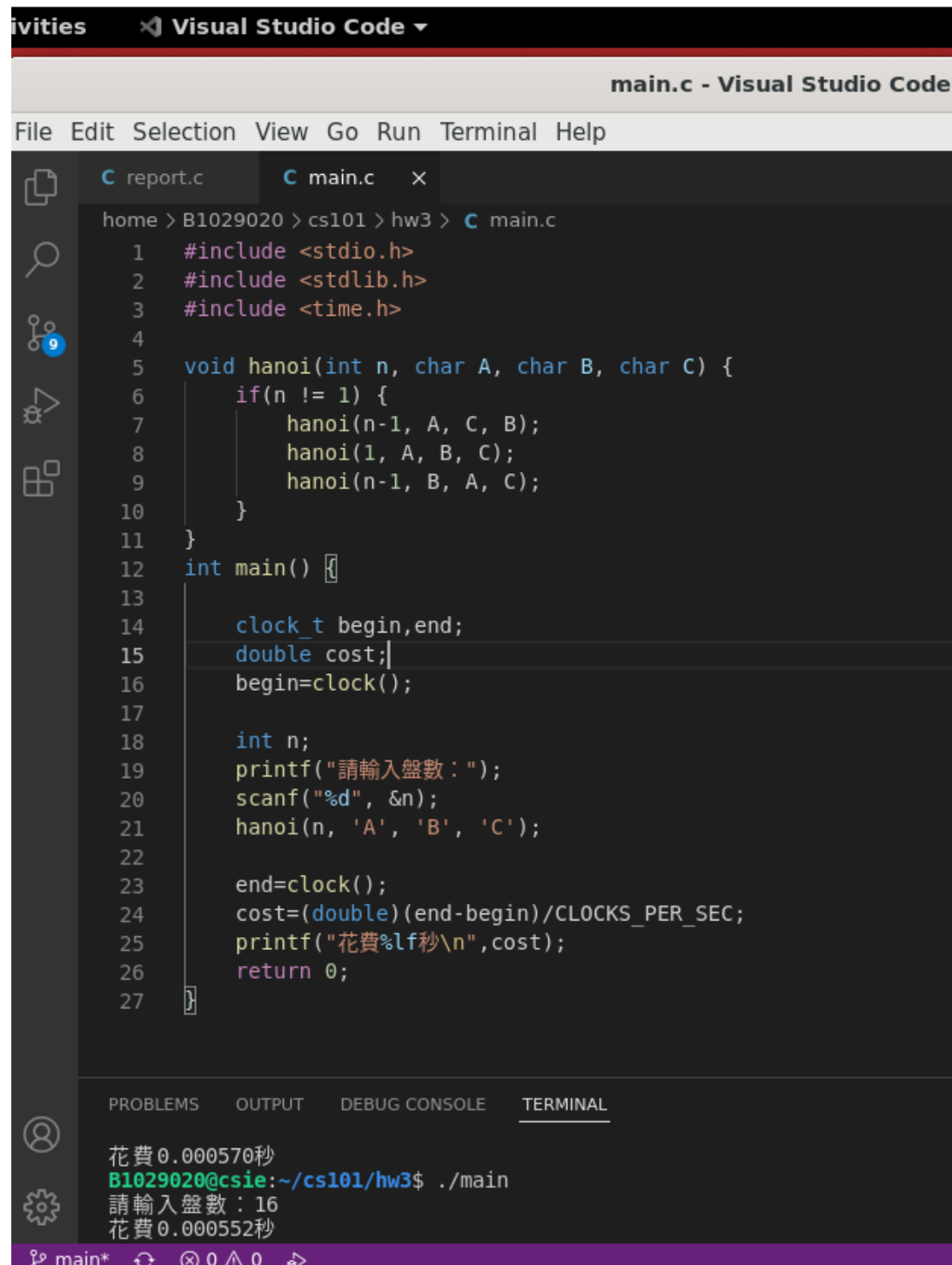
在我所撰寫的程式中我們分別設了 A、B、C 三個柱子，並把其中兩柱分別當作輔助柱以及目標柱。

`hanoi(n-1, A, C, B);`表示把輔助柱當作目標柱

`hanoi(n-1, B, A, C);`表示把原本在輔助柱的盤子搬回來

電腦的 CPU 為: intel(R) Core(TM) i5-6300HQ CPU @ 2.30GHz 2.30 GHz

20.126.17.94 - 遠端桌面連線



```
Visual Studio Code
main.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help

report.c main.c x
home > B1029020 > cs101 > hw3 > C main.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4
5  void hanoi(int n, char A, char B, char C) {
6      if(n != 1) {
7          hanoi(n-1, A, C, B);
8          hanoi(1, A, B, C);
9          hanoi(n-1, B, A, C);
10     }
11 }
12 int main() {
13     clock_t begin, end;
14     double cost;
15     begin = clock();
16
17     int n;
18     printf("請輸入盤數：");
19     scanf("%d", &n);
20     hanoi(n, 'A', 'B', 'C');
21
22     end = clock();
23     cost = (double)(end - begin) / CLOCKS_PER_SEC;
24     printf("花費%lf秒\n", cost);
25     return 0;
26 }
27

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
花費0.000570秒
B1029020@csie:~/cs101/hw3$ ./main
請輸入盤數：16
花費0.000552秒
main*
```